Project 8 Hard Spectrum Partitioning

Professor: Massimo Tornatore
Supervisor: Annalisa Morea

COMMUNICATION NETWORK DESIGN
A.A 2022/2023

Presenting:
Usevalad Milasheuski
Ignazio Maria Castrignano

Summary

–Problem
-Topology
-Spectrum
-Input
-Algorithms
-Results
-Conclusion

Table of content

- PROBLEM FORMULATION
  - HARD/SOFT SPECTRUM PARTITIONING
    - INTEGER LINEAR PROGRAMMING
    - FIRST FIT
    - GENETIC ALGORITHM
  - CREATING TOPOLOGY
  - CREATING DEMANDS

Problem Formulation

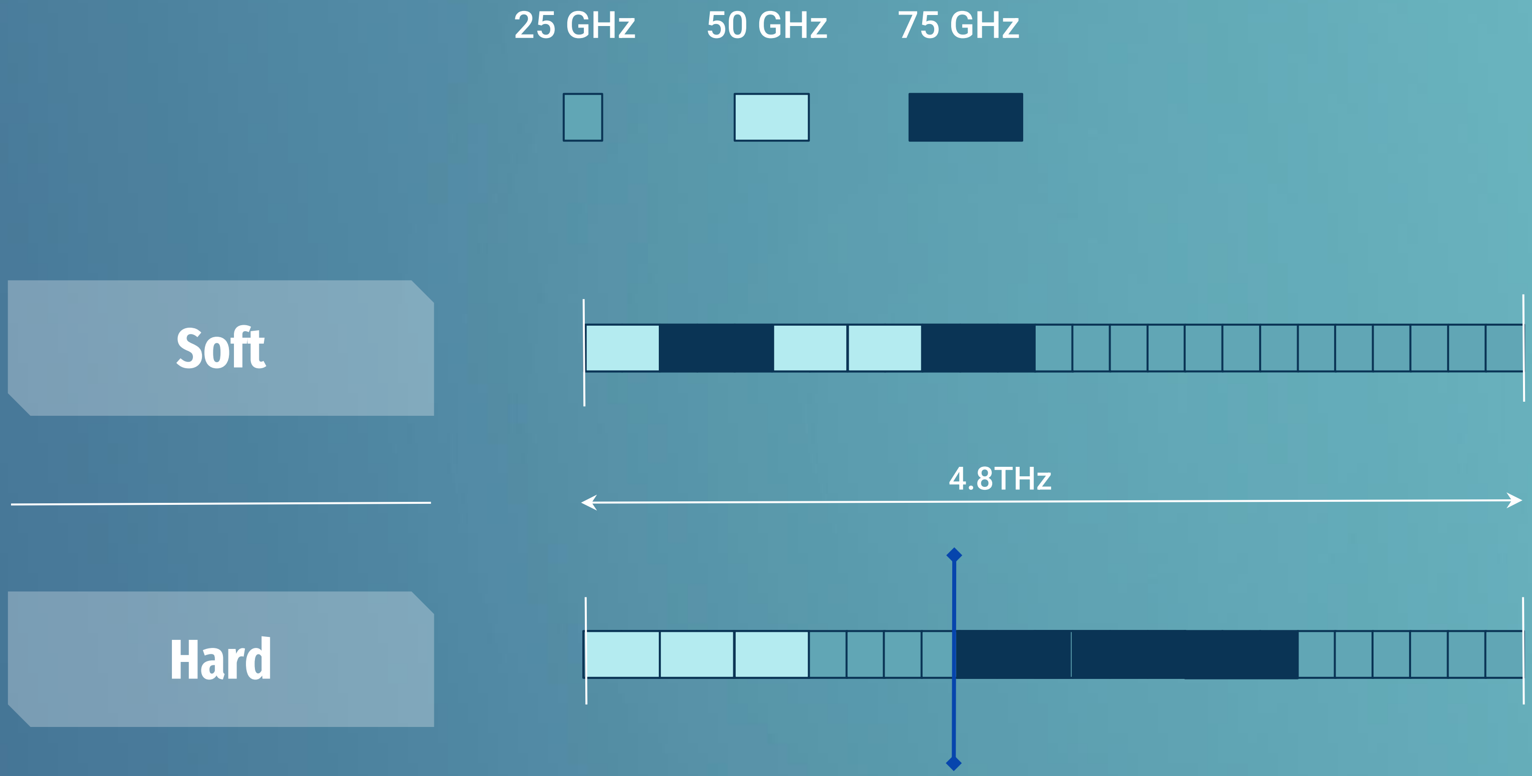
Problem

Spectral efficiency of a network with multiple channel widths and divided fiber spectrum into 2 sections

Goal

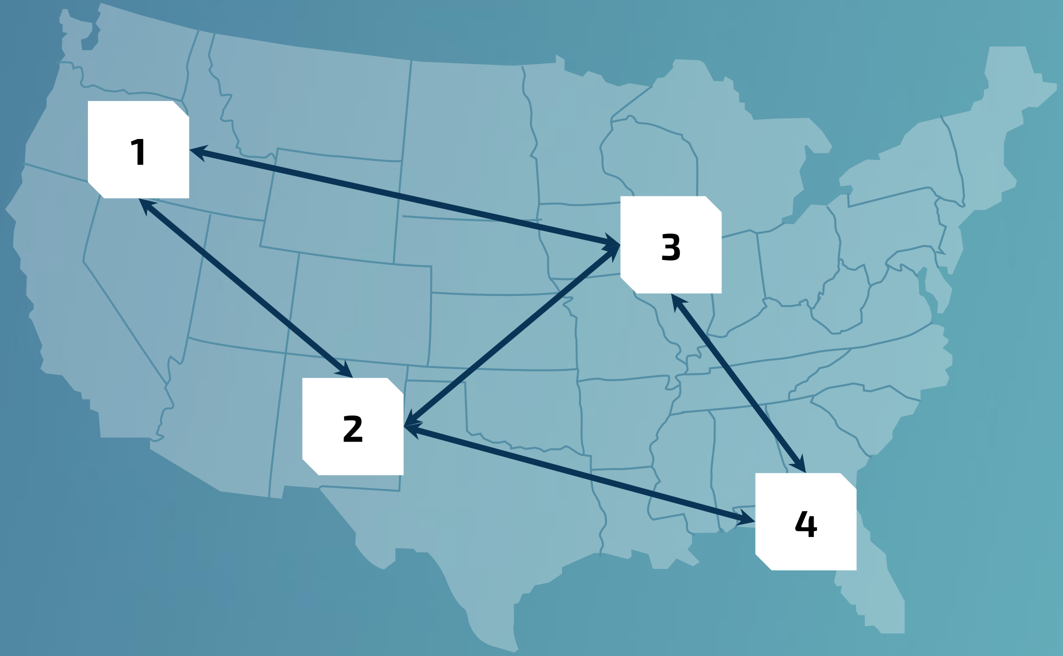
-minimize blocking probability
-minimize overall cost

Spectrum Partitioning



How Topology works

Source
Destination
Pair



K-Shortest
Path

Lenghts

Lenghts

Input to generate demands

Data Rate Gb/s	Modulation Format	Bits/symbol (Gb/s)- Entropy	Channel spacing Δf (GHz)	Reach (km)
200	16 QAM	4.00	50	900
400	16 QAM	4.00	75	600

Modulation format:

-m=0 50 Ghz 2 slices of 25
-m=1 75 Ghz 3 slices of 25

Demands:

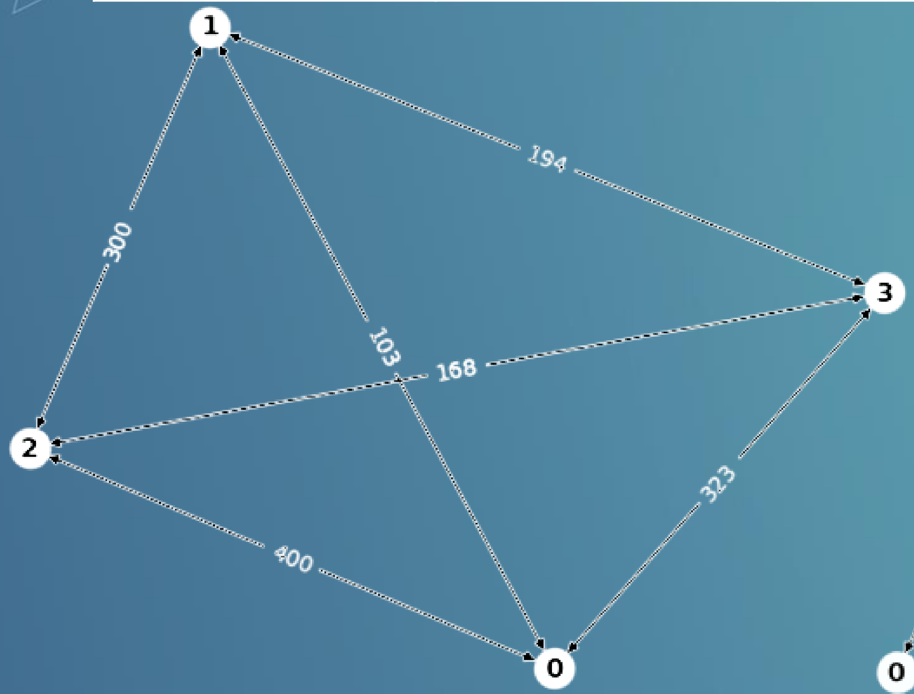
-350:5:D_max{p_blocking !>= 1%}

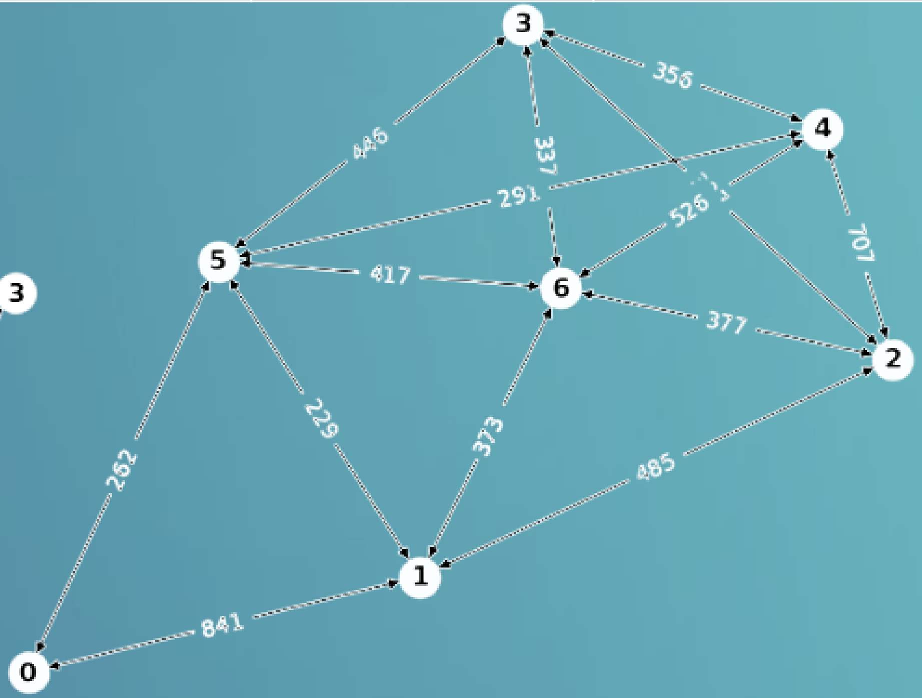
How we generate it:

```
def GenTraff(demands, Nodes):
    D = DemandList() # tuple of 3 values: s, d, b
    for d in range(demands):
        s_d = random.sample(range(0, Nodes), 2)
        D[(s_d[0], s_d[1])] = np.random.choice(
                                    [100,150,200,250,300,350,400])
    return D
```

OUR NATIONAL AND CONTINENTAL TOPOLOGIES

	Number of Links	Average link lenght	Max link length	Number of Nodes
National	≤ 20 links	~ 150 km	~400 km	4
Continental	≥ 20 links	~ 300 km	~ 1000 km	7





ALGORITHM

Integer Linear Programming

First Fit

Genetic Algorithm

ILP Variables

optimize a model with
2810 rows
153693 columns
3634301 nonzeros

Takes around 2 hours to optimize for a network with 10 links and 100 demands…

Demands and Paths:

D	Set of demands, index d. For each demand d, the tuple $\langle o_d, t_d, b_d \rangle$ is given, where o_d and t_d are the origin and target nodes, and b_d is the requested bitrate in Gb/s.
P	Set of pre-computed paths, index p.
$P(d)$	Subset of pre-computed paths for demand d. $\lvert P(d) \rvert \leq k \forall d \in D$
r_{pe}	Equal to 1 if path p uses link e.
len(p)	Length of path p computed as $\sum_{e \in E} r_{pe} \cdot \text{len}(e)$.

Spectrum and Modulation Formats:

S	Set of spectrum slices, index s.
$C(d)$	Set of pre-computed slots for demand d.
M	Set of modulation formats, index m. The reachability len(m) for every modulation format is defined.
q_{cm}	Equal to 1 if slot $c \in C(d)$ is computed for modulation format m.
q_{cs}	Equal to 1 if slot c uses slice s.

The Decision Variables are:

w_d	Binary, equal to 1 if demand d cannot be served.
x_{dpc}	Binary, equal to 1 if demand d is routed through path p and slot c.
v_{dm}	Integer equals to the number of times the demand d modulated on m has been regenerated

ILP Constrains

Using the tool gurobipy we tried to optimize the following problem adding costrains:

The AP-RMSA formulation is as follows:

$$(AP - RMSA) \min \sum_{d \in D} b_d \cdot w_d \ + \alpha \cdot \sum_{d \in D} \sum_{m \in M} 2 v_{dm}$$

subject to:

$$\sum_{p \in P(d)} \sum_{c \in C(d)} x_{dpc} + w_d = 1, \ \forall d \in D$$

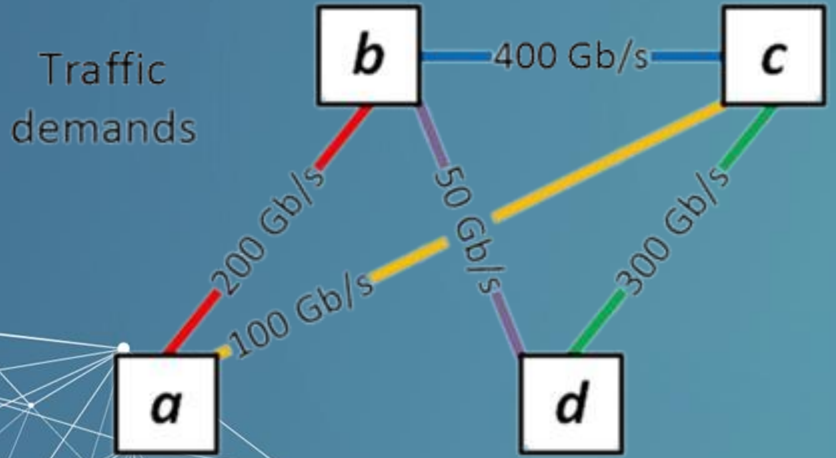
$$\sum_{p \in P(d)} \sum_{c \in C(d)} q_{cm} \cdot len(p) \cdot x_{dpc} \leq len(m) \ + v_{dm} \cdot len(m) \quad \begin{array}{l} \forall d \in D, \\ m \in M \end{array}$$

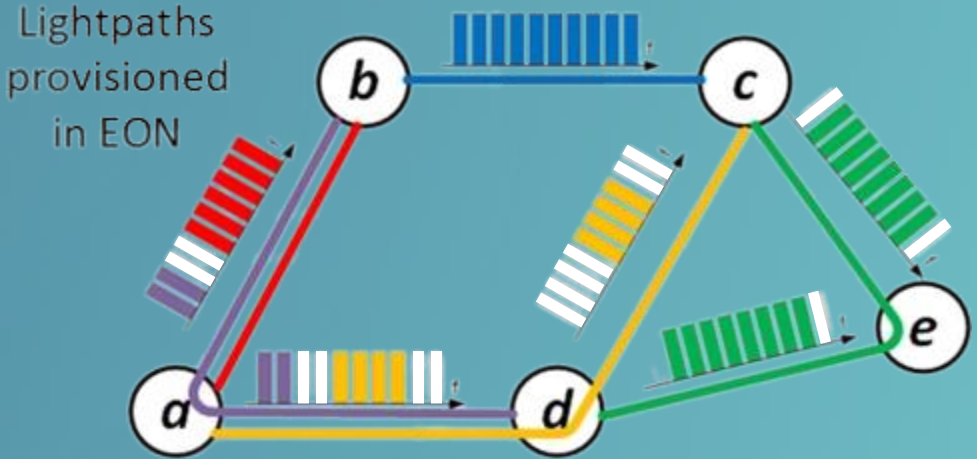
$$\sum_{d \in D} \sum_{p \in P(d)} \sum_{c \in C(d)} r_{pe} \cdot q_{cs} \cdot x_{dpc} \leq 1, \forall e \in E, s \in S$$

First Fit

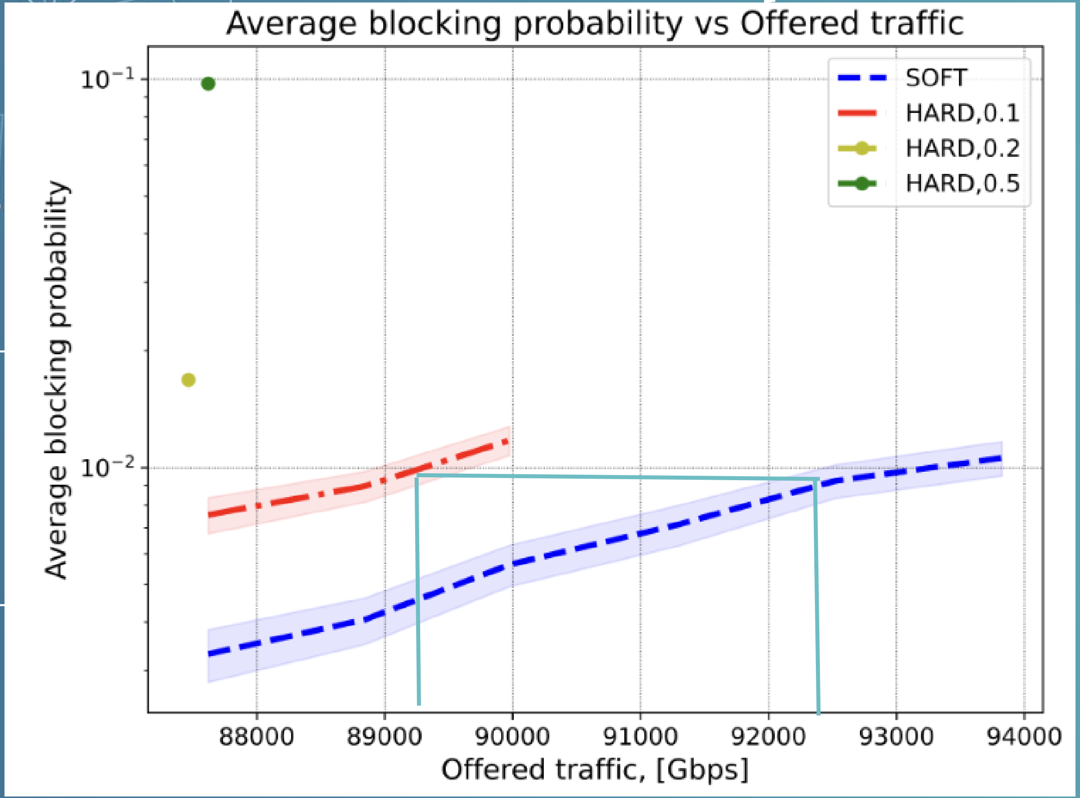
First Fit

First fit is an heuristic method.
Firstly we computes the shortest path using the K-Shortest Path.
Then take the first demand and route it in the first available shortest path, if is taken the second, and so on...





Results First Fit: National (Bp)



This is the comparison between the Soft Partitioning and the Hard Partitioning in the National Topology.

So as we can see the blocking probability using the Soft Partitioning is lower so is better for our results.

1. EX. for 89Tbps bp Soft = 2*10^-3%, bp Hard = 1%
2. EX. for 1% bp Soft we can route = 4Tbps more than with Hard

Lower Confidence Limit: $\bar{x} - 1.96\frac{\sigma}{\sqrt{n}}$

Upper Confidence Limit: $\bar{x} + 1.96\frac{\sigma}{\sqrt{n}}$

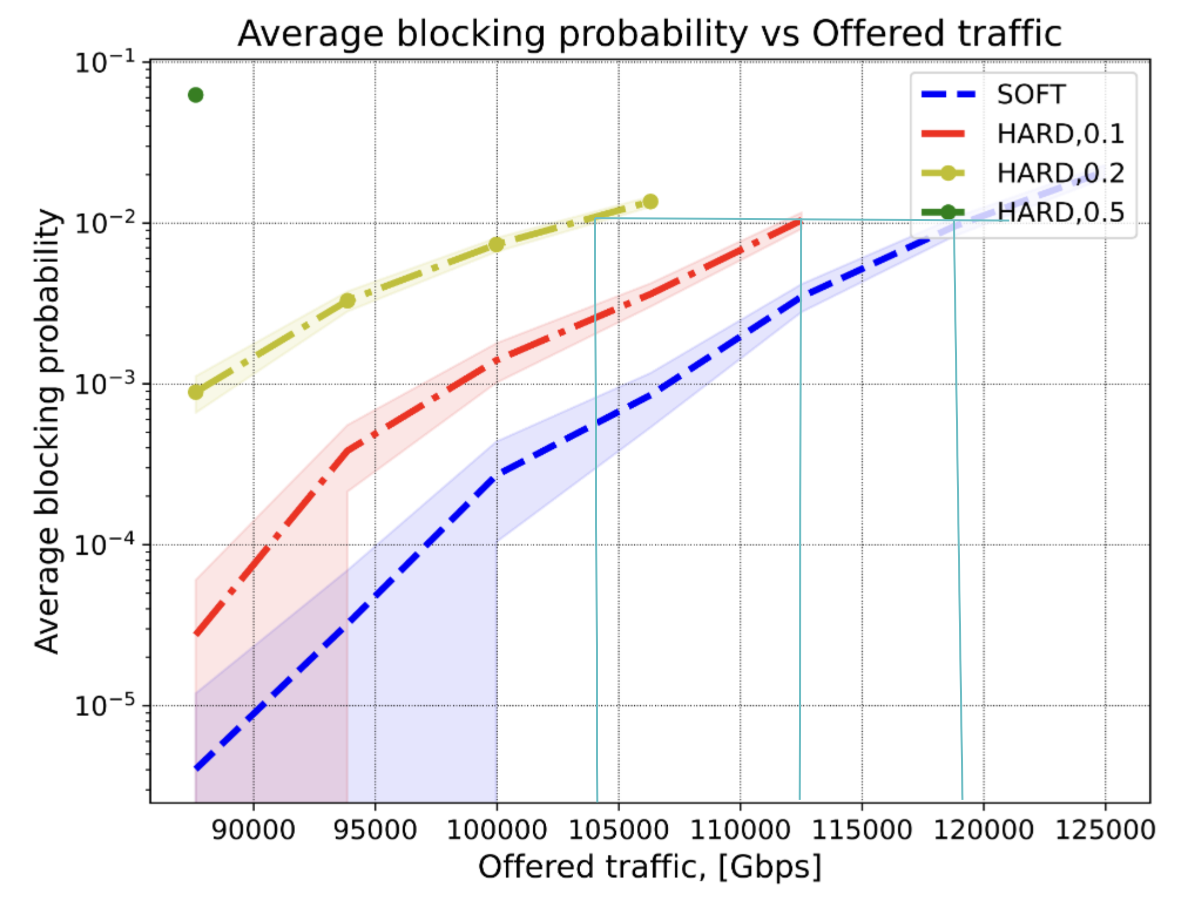
This interval of value is computed with Monte Carlo Simulation, calculating with 1000 different random demands our results 95% of times is inside this interval

Results First Fit: National (Cost)

This is the comparison between the Soft Partitioning and the Hard Partitioning in the National Topology.

So as we can see that the cost 95% of times is 0 so is negligible in the case of small topology

Results First Fit: Continental (Bp)

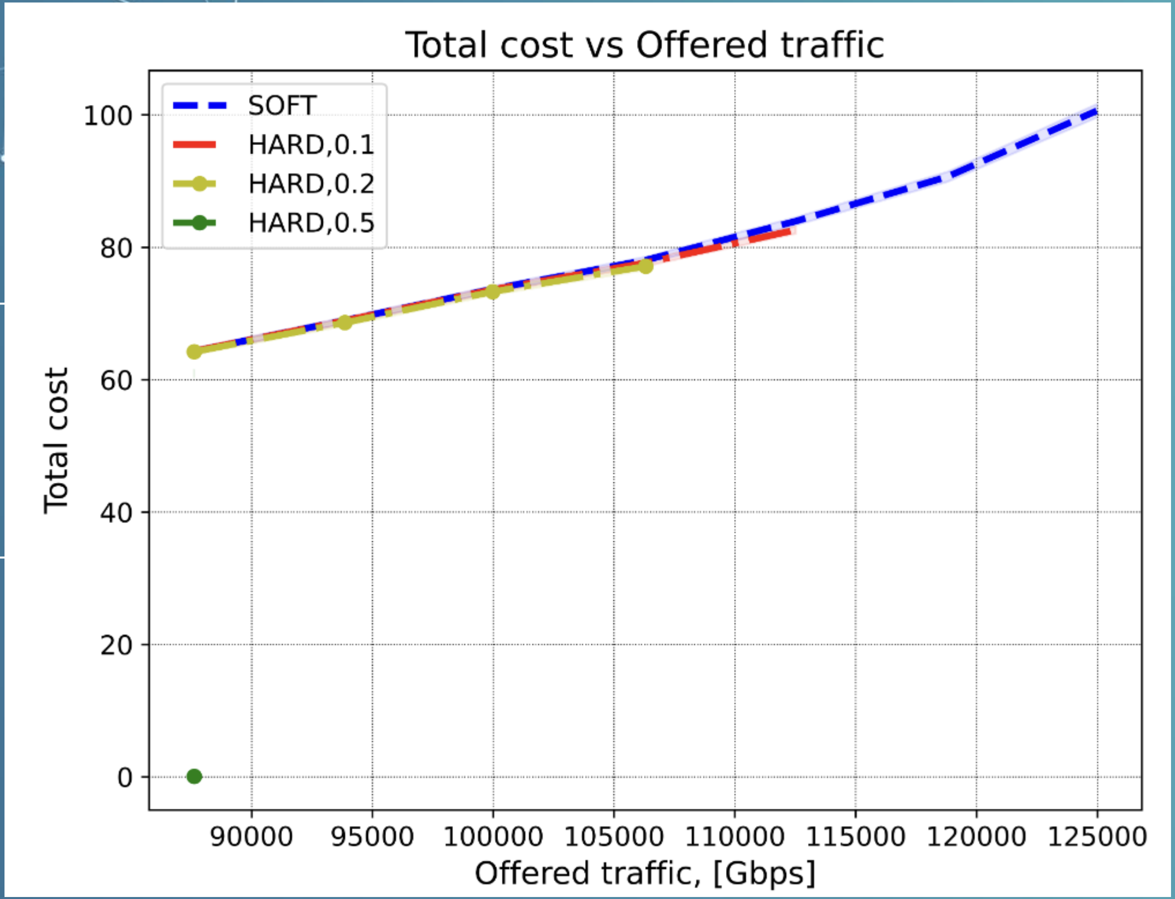


This is the comparison between the Soft Partitioning and the Hard Partitioning in the Continental Topology.

1. EX. for 95Tbps
   bp Hard,0.1 = 3,5*10^-4%
   bp Hard,0.2 = 1%
   bp Hard,0.5 = 10%

1. For the plot we can see that the lower is the border value the more traffic we can route for the same bp (not true)

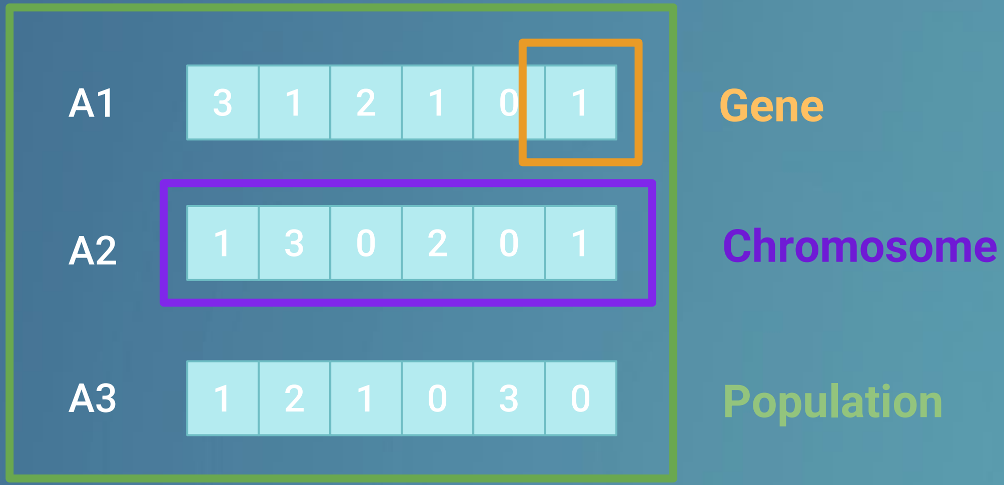
Results First Fit: Continental (Cost)



This is the comparison between the Soft Partitioning and the Hard Partitioning in the Continental Topology.

So as we can see the Total cost is very similar using the Hard or the Soft partitioning.

Genetic Algorithm

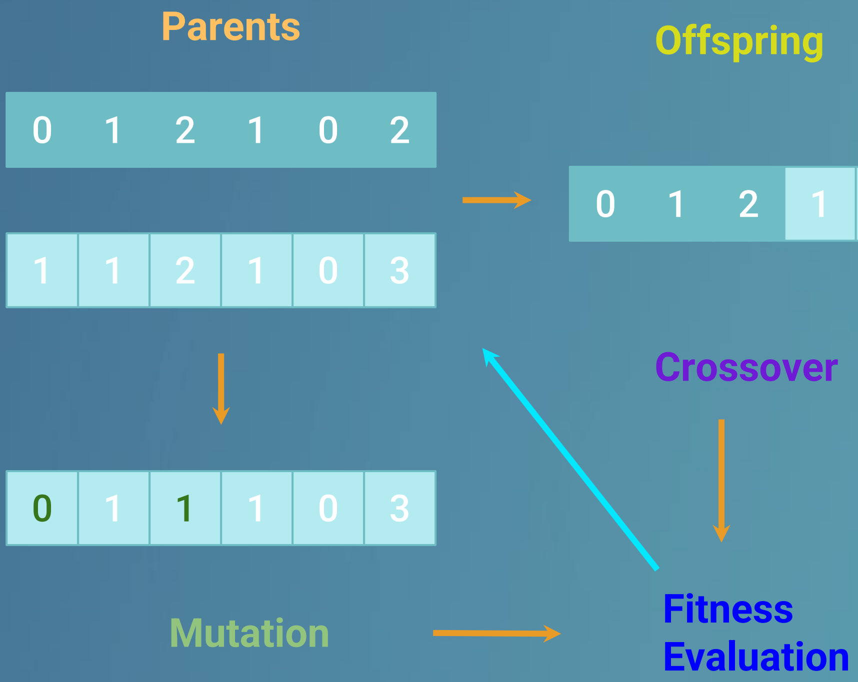


Genetic Algorithm

Is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA).
One candidate solution in Genetic Algorithm is called Chromosome.

One value of the Chromosome is called Gene.
All the calculated Chromosome are called Population.

Taking the best candidates populations as Parents, then taking the bests offrsprings and make them Parents, and so one untile find the best solution.



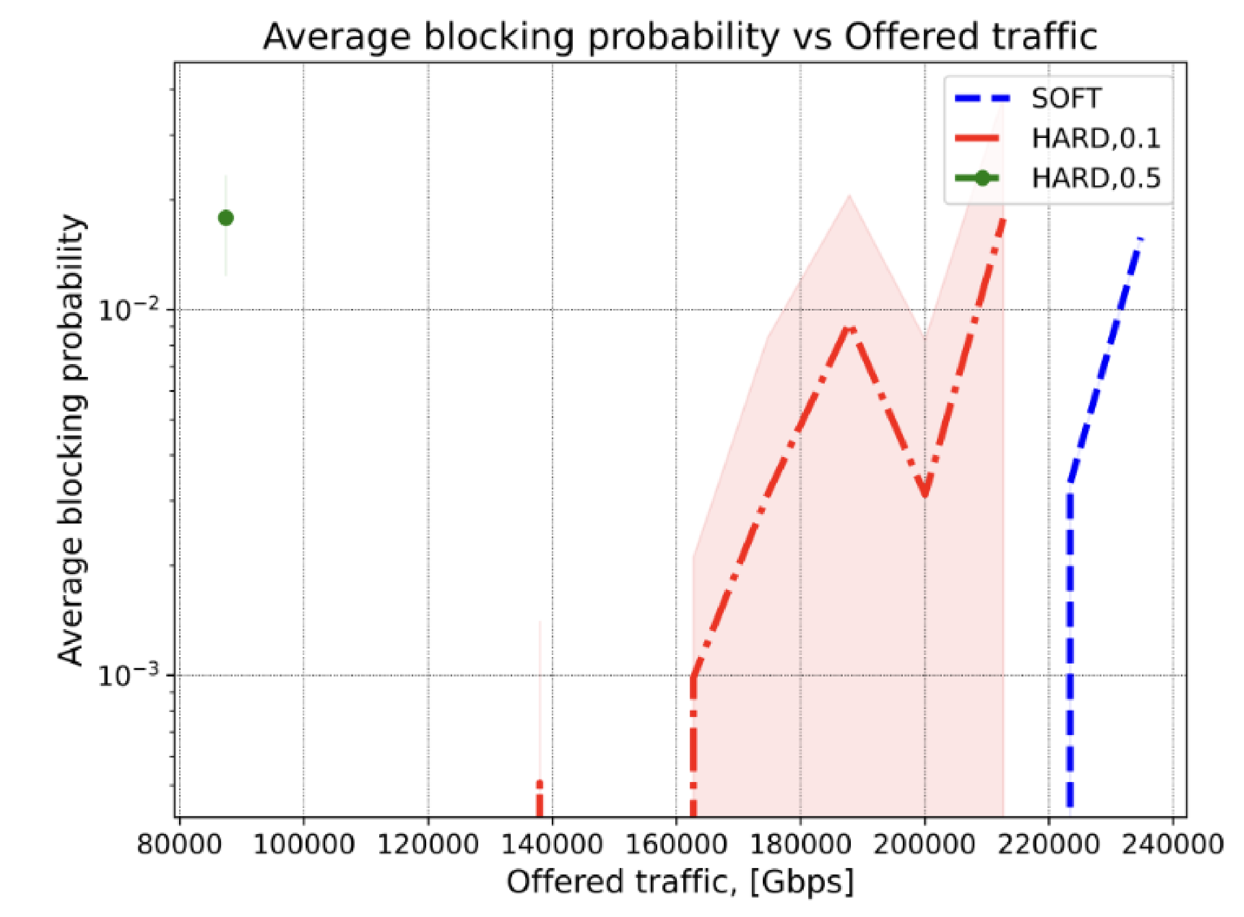
Genetic Algorithm

Genetic Algorithm take the solution using First Fit algorithm and than makes Crossover and Mutation Function.

Crossover mixes different solutions creating new candidates.
Mutation randomly changes several genes in order to adds stochasticity.

At the end evaluate the candidates and take the best ones and restart the process till the end of the tree.

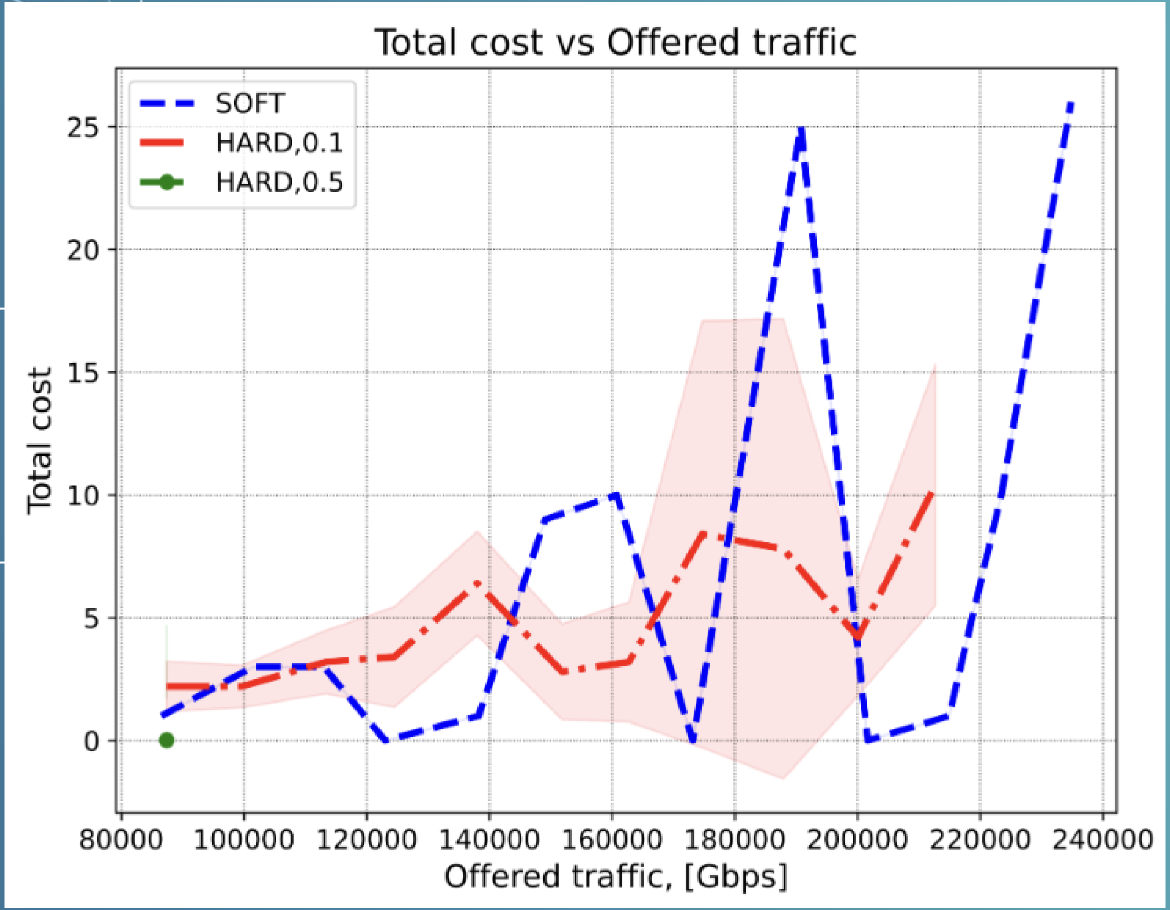
Results Genetic Algorithm : National (Bp)



This is the comparison between the Soft Partitioning and the Hard Partitioning in the National Topology.

So we can take the points as the maximum offered traffic that can be routed

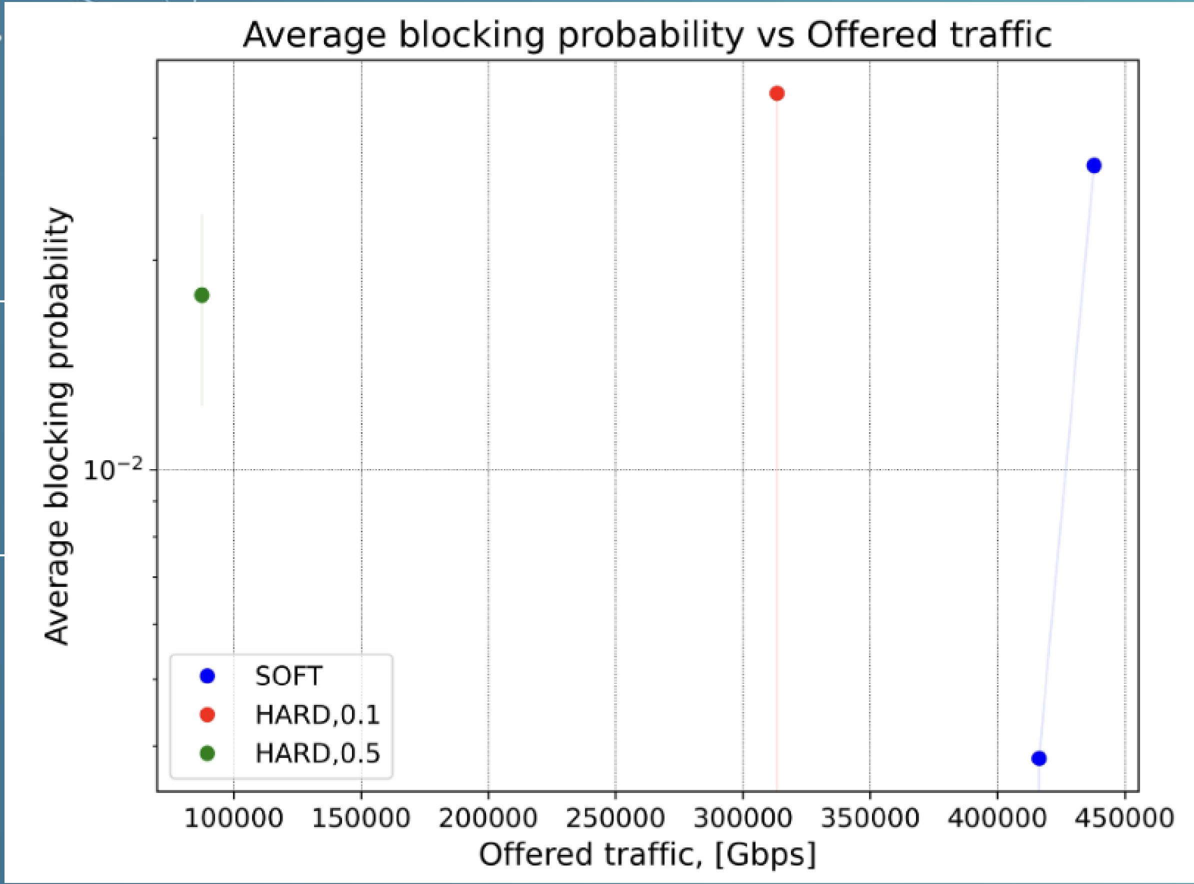
Results Genetic Algorithm : National (Cost)



This is the comparison between the Soft Partitioning and the Hard Partitioning in the National Topology.

So as we can see the Total cost is very similar using the Hard or the Soft partitioning

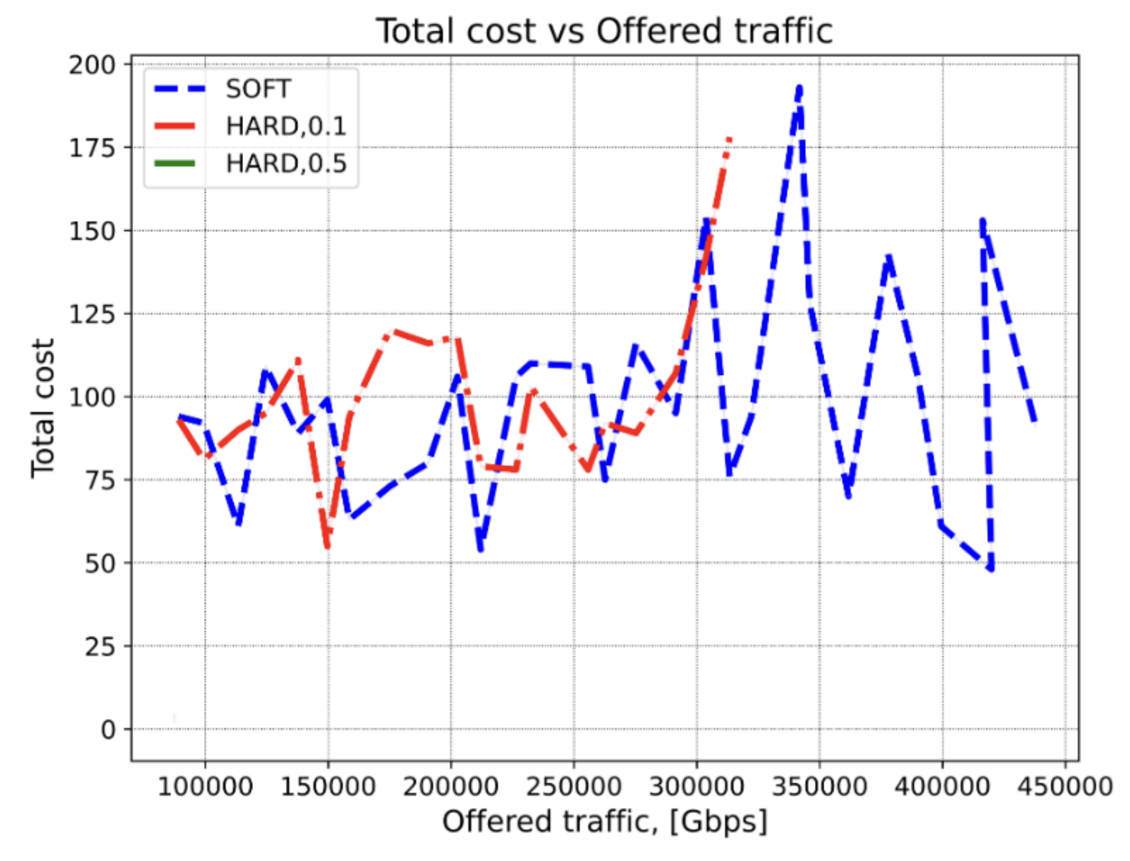
Results Genetic Algorithm : Continental (Bp)



We have only one point for each method because all other bp points are 0 or over the threshold.

So we can take the points as the maximum offered traffic that can be routed

Results Genetic Algorithm : Continental (Cost)



This is the comparison between the Soft Partitioning and the Hard Partitioning in the Continental Topology.

So as we can see the Total cost is very similar using the Hard or the Soft partitioning

CONCLUSION

1. For large number of demands ILP is not computational efficient solution.

1. Hard partitioning does not influence the cost.

1. Genetic algorithm is more effective than k-SP First Fit but it requires larger amount of time to compute.

1. (~ 2 min per demand for 1000 MC vs ~ 15 min of demands for 5 MC) .

1. Soft Partitioning may serve as a lower bound for the performance.

1. Finding best border values requires hyper-paramer search

POLITECNICO
MILANO 1863

NOKIA